Arduino Mail Notifier

Written By: Riley Porter

TOOLS:

- Pen (1)
- Wire cutter/stripper (1)

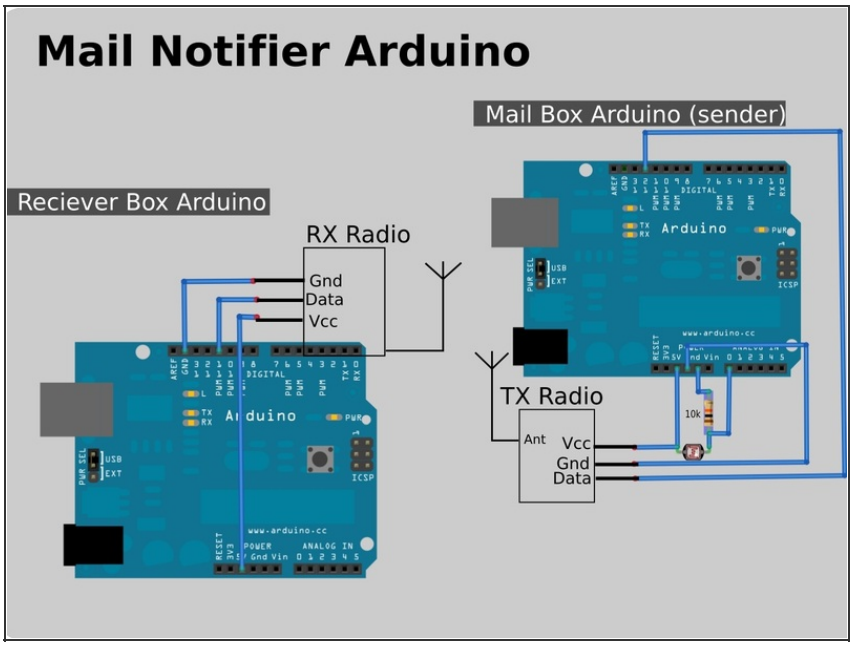
PARTS:

- Arduino microcontroller (2)
- RF Link Transmitter (1)
- Mini breadboard (2)
- 10K Resistor (1)
- Photoresistor (1)
- Battery holder, 4xAA (1)
- Breadboard Wire (1)
- RF Link Receiver (1)

SUMMARY

Many inventions/hacks often arise out of the need to solve a problem. The Arduino mailer notifier project was no exception. My workshop is in my backyard. During the day, I am far away from my mailbox. Using two Arduinos, some cheap RF modules, and a light sensor, I was able to create a solution that alerts me when my mail has arrived. In this project, I will show you how I did it.

Step 1 — Putting Together the Circuits.



- Wire up the two Arduino boards as shown in this diagram
- On the receiver there are multiple places for GND and VCC. Connect all of the GND's together and the VCC's together.

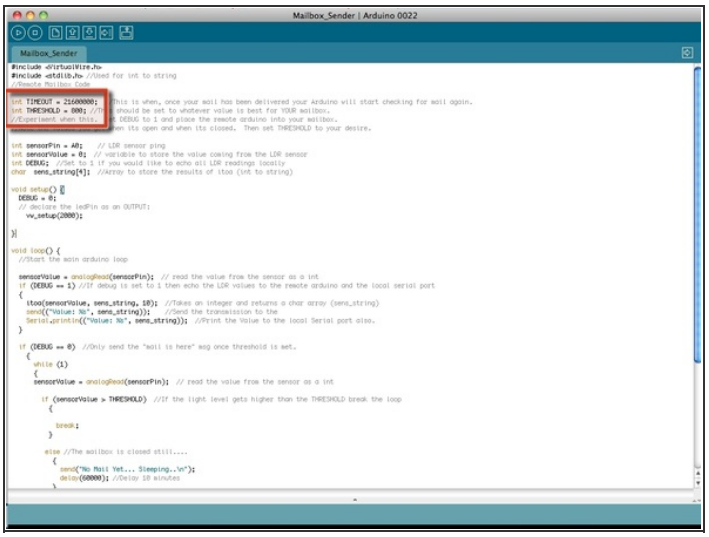
Step 2

- Build your antennas (2x)
  - Cut roughly 12" of breadboarding wire.
  - Strip 1/4" off of one side of each wire.
  - Tightly wrap one wire around a pen.
  - Remove both wires and insert them into the Ant port on each radio.

Step 3

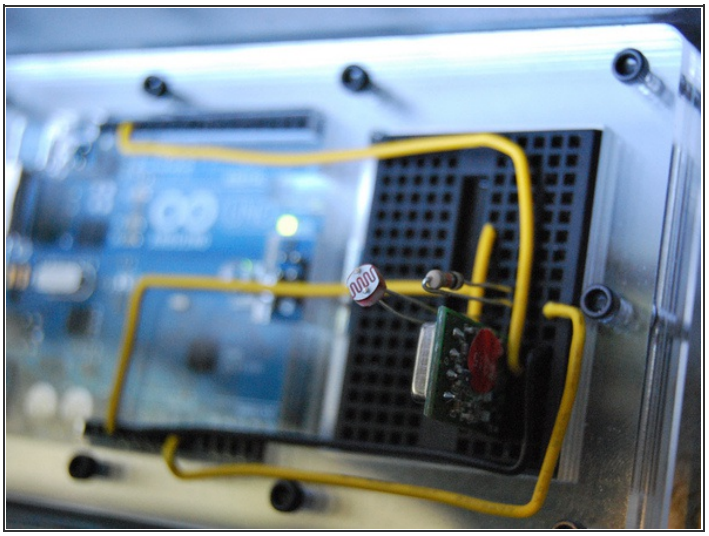
- Copy and pase the code from the URLs below into two different Arduino Sketches. Be sure to load the Receiver code on to the Receiver Arduino ciruit and vice versa.
- Receiver Code: https://github.com/ril3y/Make-Projects/r...
- Sender Code: https://github.com/ril3y/Make-Projects/r...

Step 4



- Tweak the Sender Settings.
  - The sender sketch works as follows. It continues to check to see if the LDR reading (light sensor) is above the THRESHOLD value. See the image attached. I placed a red square box around the THRESHOLD var. Since each mailbox's ambient light will vary from place to place I created this THRESHOLD variable.
  - Inside the Sender sketch is a DEBUG variable. If you set this value to equal 1. then ALL light readings will be sent to the receiver. This is a good way to see what your mail box registers while the lid is closed.
  - Record your closed lid value and your open lid value. For example: If your open lid was 300 and your closed lid was 600 then setting your THRESHOLD value to 350 might be a good value to try.

Step 5



- Test it all out.
  - Now that your Sender sketch is all tweaked for your mailbox lets see how it works.
  - Place your Sender in the mailbox.
  - Hookup your receiver to your computer.
  - Open your serial terminal window. (See image).
  - Now have someone go open your mail box.
  - If all went well you should have message that says "Got: Your Mail". It should also be noted that in the Sender sketch there is a "TIMEOUT" var. This is the number of milliseconds that will occur before your Arduino starts looking for mail again. This way you do not have to manually reset your mail checker.

This project was designed to get you used to working with cheap and simple RF modules while providing a cool, practical application. There is much room for improvement here. For instance, if this were a long-term project, much code optimization would need to be done to have the processor "sleep." Also, for alerting purposes, perhaps an LED or some audible alert could be added.

In O'Reilly's "Arduino Cookbook," in recipe 14.1 *Sending Messages Using Low-Cost Wireless Modules*, there is much more information about using these RF modules.

You can get the book here at the Maker Shed.

For another mailbox alert system, check out Matt Richardson's Snail Mail Push Alerts project.

And for lots more Arduino, take a gander at the Make: Arduino page.

This document was last generated on 2012-11-02 01:03:47 AM.